ARI Research Note 91-71

Assessment of Model Generative Reasoning for Use in the Intelligence Production Performance Model

Michael J. Coombs, Roger T. Hartley,
and Heather D. Pfeiffer
New Mexico State University

Field Unit at Fort Huachuca, Arizona
Julie A. Hopson, Chief

Systems Research Laboratory
Robin L. Keesee, Director

June 1991

United States Army
Research Institute for the Behavioral and Social Sciences

U.S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction
of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON Technical Director	JON W. BLADES COL, IN Commanding

---

Research accomplished under contract
for the Department of the Army

New Mexico State University

Technical review by

David D. Burnstein

A-1

NOTICES

DISTRIBUTION: This report has been cleared for release to the Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or the National Technical Information Service (NTIS).

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The views, opinions, and findings in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other authorized documents.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS: --

2a. SECURITY CLASSIFICATION AUTHORITY: --

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE: --

3. DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): --

5. MONITORING ORGANIZATION REPORT NUMBER(S): ARI Research Note 91-71

6a. NAME OF PERFORMING ORGANIZATION: Knowledge Systems Group, Computing Research Laboratory

6b. OFFICE SYMBOL (If applicable): --

7a. NAME OF MONITORING ORGANIZATION: U.S. Army Research Institute Field Unit at Fort Huachuca

6c. ADDRESS (City, State, and ZIP Code):
Box 30001/3CRL
New Mexico State University
Las Cruces, NM 88003-0001

7b. ADDRESS (City, State, and ZIP Code):
ATTN: PERI-SA
Fort Huachuca, AZ 85613-7000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: U.S. Army Research Institute for the Behavioral and Social Sciences

8b. OFFICE SYMBOL (If applicable): PERI-S

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: DAAL03-86-D-0001, DO 2247, TCN 90-501

8c. ADDRESS (City, State, and ZIP Code):
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62785A	790	1306	C08

11. TITLE (Include Security Classification): Assessment of Model Generative Reasoning for Use in the Intelligence Production Performance Model

12. PERSONAL AUTHOR(S): Coombs, Michael J.; Hartley, Roger T.; and Pfeiffer, Heather D. (New Mexico State University)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 90/08 TO 90/12	1991, June	26

16. SUPPLEMENTARY NOTATION: Julie A. Hopson, Contracting Officer's Representative

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Model generative reasoning
Enhanced-model generative reasoning
Intelligence analysis
Error modeling
Abduction
Cognitive modeling

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report assesses the applicability of the enhanced-Model Generative Reasoning (e-MGR) problem-solving architecture for supplying the information processing mechanisms for the Intelligence Production Performance Model (IPPM). The independent variables controlling error performance in the IPPM have analogical relationships with the operators in the e-MGR. A software demonstration illustrates the integration of e-MGR and IPPM to produce context changes resulting from errors in hypothesis generation. The e-MGR can provide a suitable set of mechanisms for augmenting the IPPM. The e-MGR mechanisms can dynamically sketch the etiology of errors and their decision efforts.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
David D. Burnstein	(602) 538-4704	PERI-SA

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ASSESSMENT OF MODEL GENERATIVE REASONING FOR USE IN THE INTELLIGENCE PRODUCTION PERFORMANCE MODEL

CONTENTS

LIST OF FIGURES

ASSESSMENT OF MODEL GENERATIVE REASONING FOR USE IN THE INTELLIGENCE PRODUCTION PERFORMANCE MODEL

Introduction

Rationale and Objectives

The best understood part of intelligence analysis is the data driven process of identifying and locating units by correlating signatures to equipment, and equipment to units, through tables of organization and equipment. However, the highest payoff comes not from simply knowing the identity and location of units, but from going beyond the unit level to identifying the global characteristics of the current situation, and from predicting enemy intentions.

Hypotheses concerning the current situation and threat intentions are valuable because they enable operations staff to anticipate future threat actions, to identify threat vulnerabilities, and to improve performance through added preparation time. Such hypotheses are, however, difficult to construct.

Intelligence products are designed to meet the decision needs of the commander with respect to a given mission. In order to achieve relevance, the analyst must go well beyond raw data to generate highly refined, mission-specific descriptions of present and future situations. Raw data concerning a complex of diverse, and often dynamic, entities, must be collected, selected, interpreted, integrated, and evaluated against both stated and anticipated commander needs. This is, not surprisingly, a difficult cognitive task. It is also ill understood and very prone to error.

Intelligence analysis is conducted in a class of task environments that may be characterized as competitive. In competitive task environments, each competitor seeks to gain control over an opponent's decisions by influencing the opponent's perception of the world. Methods of control typically include the use of (1) noise, to make it difficult for an opponent to form and test interpretations of an evolving situation; (2) deception, to make an opponent accept some desired, and disadvantageous, interpretation of situations and intentions, and (3) novel actions, which will confuse the opponent because they lie outside the explanation space. The effect is to increase uncertainty for the opposition either by casting doubt on the relevance of data (through deception and noise) or by decreasing the value of expectations (through deception and novelty).

In such environments, the analyst is in a double-bind. If the current explanation is perceived as tentative, the analyst is in danger of frequently "losing the picture." If incoherence is rationalized away through a search for confirming evidence, the analyst may drive himself into a Bayesian black hole, i.e., the more that a hypothesis is confirmed, the more evidence that will be required to disconfirm it. Moreover, difficulties become increasingly marked as behavioral constraints begin to predominate.

Folklore has identified a number of strategies that will enable an analyst to maintain control. The most frequently encountered strategy, and the one that has some experimental validation, is the maintenance of a set of alternative hypotheses in a form suitable for use as contexts for viewing available data (see Tolcott, 1989). The emphasis is on the word "alternative," since a set of hypotheses is required that spans the set of alternative operational options open to the opponent. Using this set, the analyst can: (1) generate efficient collection plans to reduce the hypothesis space; (2) anticipate alternative enemy courses of action, and (3) rapidly generate new hypotheses from the fragments of the old set to explain unexpected patterns of data.

The intelligence analysis process is sufficiently complex that it is difficult to study effectiveness analytically. However, a simulation approach requires commitment to some set of processing mechanisms. The selection of appropriate mechanisms is critical. At the very least, they must:

a. Capture domain behavior at some desired level of description.

b. Be capable of executing over data structures that are sufficiently expressive to capture significant domain input.

c. Be appropriately parameterized to allow an experimenter to meaningful control hypothesis generation.

The Army Research Institute (ARI) Field Unit, Ft. Huachuca, AZ has developed an Intelligence Production Performance Model (IPPM) as part of its program for enhancing the individual performance of intelligence staff. This model operates at a normative, information processing level, rather than at a human cognitive processing level. This is appropriate given weaknesses in our understanding of human cognition in competitive task environments, where the presence of noise, novelty, and deception are the norm.

As it stands, the IPPM is presented as a set of functional nodes, joined together in a network which process input information and pass information through an output link to the next node. Each node is represented as a black box, i.e., no particular mechanisms have been associated with the tasks carried out by each node.

The objective of this report is to assess the applicability of the Model Generative Reasoning (MGR) problem solving architecture for supplying information processing mechanisms for use in the IPPM.

Overview of the Intelligence Production Performance Model (IPPM)

The IPPM is presented as a set of functional nodes joined together in a network. Internal to each node are information processing factors believed to influence intelligence production performance at that node. Intelligence products themselves are evaluated in terms of their acceptability to an individual user, and deviations from that individual's standards are explained in terms of local "errors" occurring within particular nodes.

Input-Output Modes

The IPPM identifies several classes of independent variables Information processing at the nodes are influenced by these variables.

Information State. The Information State constitutes the information (combat information, processed data, or intelligence) which must be used to produce the final intelligence output. It is measured in terms of five dimensions:

The amount of information contained.

The relevance of information to a given node function.

The variety of types of information contained.

The spatial or temporal configuration of information.

The complexity of information.

Control State. Control State variables include factors externally imposed on processing, for example, as the mission, that provide processing goals, or operational idiosyncracies that constrain processing (e.g., that focus attention on, or distract attention from specific information).

Task State. Task State variables define the task situation within which an operator must perform. They include variables which affect task performance, for example, task difficulty, the time allowed for performance, workload.

Operator State. Operator State variables define the cognitive content and procedural knowledge the operator brings to the task, as well as any physiological states.

Performance Criteria

Final intelligence production performance in the model is defined in terms of the acceptability to some given intelligence product to a given user (Burnstein, Fichtl, Landee-Thompson, & Thompson, 1990). Five criteria are used to define "acceptability."

Completeness: e.g., who, what, when, where, why, and how?

Operational Perspective: how well an information item was put in the context of current or future friendly force operations.

Clarity: how easily content was understood or followed by the user.

Timeliness: whether the item was received in time for the user to take action.

Frequency: how often an item is provided to keep the user fully-informed.

System "Errors"

Deviations of output from the user defined product are explained in terms of "errors" originating within the nodes of the model. In the current state of IPPM development, an error taxonomy of six behavioral categories has been defined. Classes of error include the following:

> Complying with the control state: Errors related to the existing administrative constraints, directions, or guidance.
>
> Collecting the information from the environment: Errors related to collecting information necessary to perform the task.
>
> Recalling cognitive knowledge: Errors related to declarative and procedural knowledge recall.
>
> Executing the procedures: Errors related to:

Assessment of the information state given the control and operator state.

Formulation of hypotheses based on assessment.

Generation of predictors based on hypotheses.

Hypothesis reformulation or refinement.

Hypothesis testing: Errors relative to refuting or verifying predictions.

Selecting hypotheses: Errors related to selection of hypothesis information.

Within each category, generic errors are identified.

The Model Generative Reasoning (MGR) Architecture

Informal Overview

The MGR architecture was developed in the Computing Research Laboratory (CRL), New Mexico State University to support problem solving in competitive task environments (Coombs & Hartley, 1987; 1988; Coombs et al., 1990). In particular, it was designed to accommodate a variety of control mechanisms required for coping with noisy data, and novel situations. This architecture has evolved into the formal evolutionary-Model Generative Reasoning (e-MGR) system. This system allows manipulation of hypotheses at a higher level by using a simplified representation at its base. The e-MGR will be embedded in the IPPM.

Problem solving in e-MGR is implemented through a process of building sets of hypothetical conceptual structures to explain the concepts in available data. Since all objects in e-MGR are represented as graphs, it is possible to define "explanation" in terms of set relations between concept nodes in the graphs representing data and concept nodes in the graphs representing knowledge; more specifically, in terms of the set covering relation between data concepts and knowledge concepts; in e-MGR pre-defined knowledge structures are termed definitions, data are termed facts, and explanatory hypotheses are termed models.

In this respect, problem solving in e-MGR is related to the generalized set covering view of abductive problem solving developed by Reggia et al. (1985) where, given data, the task is to find the best set of hypotheses to explain the data in terms of the most parsimonious cover of the data by this set. However,

whereas generalized set covering deals with atomic explanatory hypotheses and pre-defined relevance relations between hypotheses and data, the requirement that e-MGR should function in noisy and novel task environments makes it necessary for the system to be capable of: (1) creating hypotheses autonomously from knowledge fragments, and (2) autonomously identifying relevant data from the set of available observations.

Hypotheses, e-MGR models, are generated through a set of graph transformation operations: (1) specialize, which builds new graphs from graph fragments, "gluing" together facts with definitional material to generate models; (2) fragment, which decomposes graphs into fragments, "ungluing" models to extract fragments worth preserving as assumptions to be passed on to subsequent stages of processing, and (3) classify, which tags a graph with a pre-computed marker graph, using assumptions to tag new facts to be submitted for processing. The critical problem solving notions here are that: (1) e-MGR interprets facts by gluing them together with definitional material to form models; (2) models are unglued to form assumptions (proto-facts) and (3) assumptions are used to extract new facts from the world, which then become interpreted to form new, more complete models.

Formal Overview of e-MGR

The e-MGR is logically a multi-instruction, multi-data parallel virtual machine (MIMD) that accepts input from the databases F and D. F is a fact database that receives all input from external agents; D is a definition database that contains all of the system's pre-computed explanations, and serve as an initial set of hypotheses concerning the relatedness of facts. The output from specialize is a database of models, M, that contains explanations currently under development. These are then input to the fragment operator and new hypotheses are produced as models. These models may then be re-entered into the system as assumptions, A. Assumptions may: (1) be constructs that help select new factual information (see C1 below); or (2) contain definitional information to be used in new covers by specialize. A data flow diagram of the e-MGR architecture is given in Figure 1. Detailed description of the lower-level operators join, **J**, cover, **C**, project, **P** and uncover, **UC**, are given in Hartley and Coombs (1989). Informally, **C** identifies a subset of definition graphs that has some pre-defined set cover relation to all of the labeled nodes in a given subset of graphs; **J** merges two graphs at a single point where both graphs contain related node labels; **P** is the inverse of join in that it seeks to identify related labels between graphs; **UC** is the inverse of cover in that it partitions graphs in the neighborhood of subgraph boundaries.

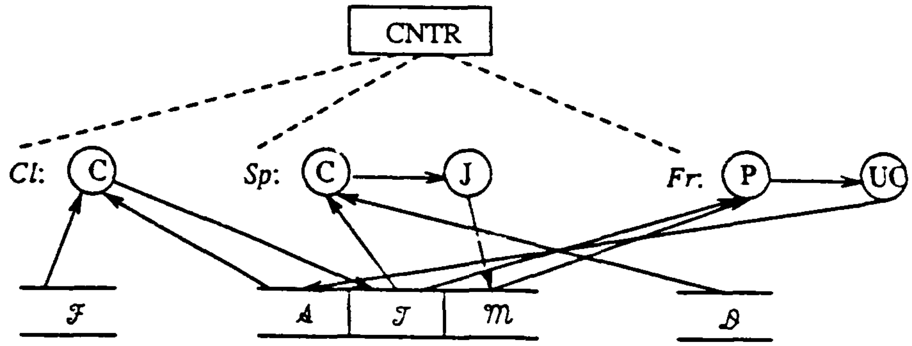

Figure 1. A data-flow diagram of the e-MGR architecture.

Three operators, classify, Cl, specialize, Sp, and fragment, Fr act on the databases in an autonomous fashion. The functionality of these operators is specified completely by the architecture. Operator actions may be described informally as follows: (1) Cl selects tagged facts T for interpretation from processing of A and F; (2) Sp generates model M interpretations by fusing items from T using definitional "glue" taken from D, and (3) Fr generates new assumptions by cleaving models through the removal of "glue" around the items currently in T. The e-MGR operations can be represented as a closely coupled set of functions, with coupling at T, M, and A. In the worst case:

Classification
$$Cl: \quad A \times 2^{F} \dashrightarrow 2^{T}$$

Specialization
$$Sp: \quad 2^{T} \times 2^{D} \dashrightarrow 2^{M}$$

Fragmentation
$$Fr: \quad M \times 2^{T} \dashrightarrow 2^{A}$$

The activity of the operators is governed by the control level, which determines when the operators act, but not their functionality. Strategy in e-MGR thus consists largely of scheduling these three operators, along with the additional activities of selection over F and D, and evaluation of A and M in order to determine halting conditions. Control strategies are formally optimizations, represented either as algorithms or adaptive systems.

Academic Connections

The three e-MGR operations can be interpreted in terms of Pierce's (Pierce, 1934) explanation cycle \(-> induction \(-> abduction \(-> deduction \(->. Classify implements the induction

of relevance relations between assumptions and facts, by which facts are selected to be considered for integration in the next round of hypothesis building. Specialize implements the abduction of interpretive contexts for tagged facts. Fragment, on the other hand, implements the deductive evaluation of hypotheses to create new assumptions from models in order to focus the next round of interpretation.

It can be seen that e-MGR moves beyond the current agenda of artificial intelligence (AI) in its study of automated reasoning to establish logic as the foundation for inference in intelligent systems (c.f., Charniak, 1986; Hanks & McDermott, 1986; Hayes, 1985; McCarthy, 1980; Shoham, 1988). The deductive view arises from the assumption that human reasoning is best characterized as deduction. The formalization of the deductive component of inferential behavior thus becomes a necessary precondition for understanding intelligent systems.

The counter argument that many inferences are not deductive has been made both in response to the difficulty of doing AI with predicate logic (e.g., McDermott, 1987), and as a belief held by those who argue that, even if intelligence could be described deductively, the critical axioms would only emerge from a prior understanding of the mechanisms of reasoning (e.g., Minsky, 1985). The difficulty of formalizing such inferential forms as abduction and induction, at least at the level of complexity under-taken by human reasoners, is typically quoted as evidence that there is more to reasoning than deduction. However, the debate has largely ended here. As McDermott has noted (1986) with reference to abduction, it is not possible to explore the relation between logic and non-deductive reasoning without a well-defined account of the non-deductive form. More particularly, a method is required to link the syntax of logical inference with the largely unformalized, semantic level of description used for representing abduction.[1]

The objective of the MGR project in general is to devise a well-defined architecture that provides a small number of mechanisms for establishing and preserving the pre-defined relational properties of a representational syntax and for relating these in a principled manner to the semantics of explanation and coherence (Coombs & Hartley, 1987; 1988). In contrast to other related work in artificial intelligence, including the ATMS methodology (de Kleer & Williams, 1987) and the

[1]Our use of the word "semantic" is important because we intend to show that abductive reasoning can be represented in terms of well-defined operators which combine purely syntactic operations on knowledge structures with semantic notions of relevance and adequacy.

explicit representation of control in expert systems, MGR seeks: (1) to describe both the non-logical domain-specific[2] aspects of problem solving and the management of alternative viewpoints in the same formalism, and (2) to describe and formalize control in terms of measures of structural transformation, rather than at the knowledge level or the calculus level.[3]

The focus of current e-MGR work is abduction, rather than induction, i.e., on the creation of structures to use in the selection of data, rather than on the role of data in creating new selective structures. This is because of the social structure of intelligence analysis, with its emphasis on the development of mission related products from available data, highlights interpretation rather than perception.

That e-MGR explanations are truly abductive, and will contain information of a hypothetical nature (i.e., that is not contained in the facts), is evident from the operation of the primitive procedures cover and uncover that implement the gluing and ungluing of graphs. Cover interprets tagged facts by first finding some subset of definitions that subsume the facts, and then fusing facts and definitions by coalescing on common concepts. The resulting explanations will therefore contain facts joined by non-factual material. Uncover cleaves an explanation into one or more fragments around the images of facts projected onto it by removing links between projections. Links may not necessarily be cut exactly at projection boundaries, thus leaving nodes that originate from definitions attached to the fragments. Uncover is not simply the inverse of cover.

The Integration of e-MGR with the IPPM

IPPM/e-MGR Relationships

The following analogical relationships have been identified between the IPPM variables and parts of the e-MGR architecture.

Information State (IS). The IS corresponds to the fact database used by e-MGR. The e-MGR assumes that facts (observations, intelligence reports) are passed to it. These facts include the set of current and past propositions about the world. The e-MGR

[2]For instance, de Kleer and Williams (1987) mention heuristics and other non-logical relationships such as the management of reasoning under uncertainty.

[3]The claim we are making is that control resides in a level of abstractions intermediate between the calculus and knowledge levels. It is thus independent of the domain and also of any knowledge representation scheme. This is the level of the operators in MGR.

also assumes that any concept identified in the input data is present in the knowledge base of the system.

Control State (CS). The CS corresponds broadly to the schema database in e-MGR. A schema is a very flexible method of representing everything from static relationships between objects, to procedures and processes that employ objects. There can be multiple schemata for any one concept corresponding to different viewpoints (i.e., opinions, strategies, personal idiosyncracies).

The adaptation of e-MGR to IPPM will require mechanisms to impose some order in which schema may be processed. This will be necessary to ensure that mission schemata are taken before doctrinal schemata, and may damp possibilities for fragmenting mission statements. Focus of attention, or switches in attention, may also be implemented in terms of schema priorities.

Task State (TS). The TS corresponds to those elements of the high-level e-MGR algorithm (see Operator State) concerned with the management of resources. The e-MGR is a very computationally expensive process. In fact, any abductive procedure has been shown to be NP-complete, i.e., exponential in the essential parameters. The e-MGR has therefore to marshal its resources carefully and monitor its own progress so as not to exceed the limitations of the machine it is running on.

Operator State (OS). The OS corresponds to the high-level algorithm used to drive e-MGR. With the current system, every application has a hand-crafted algorithm that contains an algorithmic embodiment of the goal, or goals, that make choices appropriate to the pragmatic constraints of the domain. A special purpose language will be used to specify the input to a parameterized version of e-MGR. The values of parameters may either be held constant throughout a run, or be varied under feedback.

The Demonstration Software

Overview

A Low Intensity Conflict (LIC) scenario served to illustrate the integration of MGR and the IPPM. A fictitious scenario was developed and is reported on in detail in another document (Coombs, 1991).

The Hunch Buddy Domain

The software developed to demonstrate e-MGR in this setting is configured as a decision aid called the "Hunch Buddy." The essential purpose of such an aid is to give its user the following capabilities:

a. To create and maintain a data base of factual information such as would be obtained from intelligence reports and from data analysis programs such as telephone toll analysis, link and pattern analysis, or database searches. This is the fact database.

b. To create and maintain a knowledge base of schematic structure representing the base knowledge of the user in chunked form. This is the control state.

c. To generate hypotheses abductively from selected facts in the fact database by covering them with appropriate schemata from the knowledge base. The algorithm for doing this corresponds to the operator state.

d. To display the results of the abduction to the user, the facts in the fact database and the schemata in the knowledge base.

e. To enable the user to select new facts in another cycle, to be used with previous hypotheses, until satisfactory results are obtained.

The central schema in the knowledge base concerns an insurgency drug ring conspiracy and the roles within it. The schema connects a FIXER as a central player, while a COURIER, a RECEIVER, a WHOLESALER and a FINANCIER are connected in a network with him. The purpose of the conspiracy is to gain money to support terrorism. Other schemata concern the linkage in pairs of these roles, and the identification of the roles from supporting evidence such as place of employment.

The scenario also models the piecemeal pattern of data collection, i.e., the data is not all available instantly, but either arrives over a period of time, or is the result of data collection activities based on the prior generation of good hypotheses.

Modifications to e-MGR

The full MGR software was developed on the Symbolics and is written in Common Lisp. A much cut-down version, which made many simplifying assumptions, written in C and runs on any UNIX system. It was decided to augment the C version to bring it sufficiently close to the full version so that the Hunch Buddy would demonstrate the successful completion of the task. In order to do this, several additions had to be made. These were:

a. To allow the knowledge base to have more than 32 different concept types. The e-MGR now allows up to 64 (the full version allows unlimited types).

b. To add a hierarchy of types to allow graphs to join on maximal common subtypes of two concept types, not only on the identical type.

c. To add a mechanism to simulate the repetition of type labels within a single graph. This involves manipulation of the type hierarchy to provide multiple subtypes where necessary by the addition of a suffix digit, e.g., PERSON1, PERSON2, etc., and to modify the join algorithm to simulate the multiple join possibilities of the full version.

d. To add a database system to allow interactive input of raw data and to process this data in a variety of ways to produce fact graphs for input to the abductive phase.

e. C and D give e-MGR the flexibility of representing knowledge at the most appropriate level and remove a severe restriction from the original e-MGR. In addition, they provide for a good deal of expansion capability for the future. None of the changes were specific to the LIC domain or the Hunch Buddy. All are generic additions to either the problem solving capability of e-MGR or to its capability to accept data from any source. Indeed, the database facility, albeit simple, is something that the full version of MGR lacks.

The Demonstration Data

The Database

Below is a table showing the content of the database. Each entry is self-explanatory, except for the entries with 'CALLS' in them. Each of these is assumed to be the conclusion of a telephone toll analysis program and is the single entry made in the database resulting form the analysis of possibly hundreds of telephone calls. All other entries come from direct reports of various sorts.

Cycle #	Item 1	Relation	Item 2	Certainty	Date
1	COURIER	IS	Chavez	90	07-05-88
1	Chavez	WORKS	Mort-Mex	100	07-05-88
1	IMPORTING	BUSINESS	Mort-Mex	100	07-05-88
1	Morton	OWNER	Mort-Mex	100	07-05-88
1	ORGANIZATION	IS	Mort-Mex	100	07-05-88
1	PERSON	IS	Morton	100	07-05-88
2	IMPORTING	BUSINESS	Baroni	100	07-05-88
2	Morton	CALLS	Ramon	75	09-10-88
2	ORGANIZATION	IS	Baroni	100	07-05-88
2	PERSON	IS	Ramon	100	07-05-88
2	Ramon	OWNER	Baroni	100	07-05-88
3	DISTRIBUTOR	IS	Doug	80	11-03-88
3	DISTRIBUTOR	IS	Simpson	80	11-03-88
3	PERSON	IS	Smith	100	11-03-88
3	Ramon	CALLS	Boley	60	11-03-88
3	Ramon	CALLS	Smith	40	11-03-88
3	WHOLESALER	IS	Boley	80	11-03-88
4	Broder	OWNER	Sanders	100	11-12-88
4	FINANCE	BUSINESS	Sanders	100	11-12-88
4	Harvey	WORKS	Sanders	100	11-12-88
4	Morton	CALLS	Harvey	65	11-12-88
4	ORGANIZATION	IS	Sanders	100	11-12-88
4	PERSON	IS	Broder	90	11-12-88
4	PERSON	IS	Harvey	75	11-03-88
6	Evans	CALLS	Sanders	80	01-5-89
6	Evans	OWNER	Gosling	100	01-5-89
6	INSURANCE	BUSINESS	Gosling	100	01-5-89
6	ORGANIZATION	IS	Gosling	100	01-5-89
6	PERSON	IS	Evans	100	01-5-89

The Knowledge Base

Below are the schemata in e-MGR's knowledge base. Each one contains the following items:

a. The type of schema (its 'Cast').

b. A measure of its importance (its 'Weight').

c. An identifying label (its 'Name').

d. The links in the schema between its constituent types (its 'Arcs').

Each arc links two labels. The whole set of arcs makes a graph.

```
Graph {
     Cast Definition;
     Weight 25;
     Name FIXER;
     Arcs
     FIXER -> ORGANIZATION,
     ORGANIZATION -> IMPORTING;
}

Graph {
     Cast Definition;
     Weight 25;
     Name LAUNDERER;
     Arcs
     LAUNDERER -> ORGANIZATION,
     ORGANIZATION -> FINANCE;
}

Graph {
     Cast Definition;
     Weight 25;
     Name LAUNDERER;
     Arcs
     LAUNDERER -> ORGANIZATION,
     ORGANIZATION -> FINANCE;
}

Graph {
     Cast Definition;
     Weight 25;
     Name RECEIVER;
     Arcs
     RECEIVER -> ORGANIZATION,
     ORGANIZATION -> IMPORTING;
}

Graph {
     Cast Definition;
     Weight 25;
     Name COURIER;
     Arcs
     ORGANIZATION -> COURIER,
     ORGANIZATION -> IMPORTING;
}

Graph {
     Cast Definition;
     Weight 25;
     Name WHOLESALER;
     Arcs
     WHOLESALER -> ORGANIZATION,
     ORGANIZATION -> BUSINESS;
}
```

```
Graph {
     Cast Definition;
     Weight 25;
     Name FINANCIER;
     Arcs
     FINANCIER -> ORGANIZATION,
     ORGANIZATION -> FINANCE;
}

Graph {
     Cast Definition;
     Weight 25;
     Name CONSPIRACY;
     Arcs
     FIXER -> FINANCIER,
     FIXER -> COURIER,
     FIXER -> LAUNDERER,
     FIXER -> RECEIVER,
     WHOLESALER -> LAUNDERER,
     WHOLESALER -> RECEIVER,
     WHOLESALER -> DISTRIBUTOR;
}

Graph {
     Cast Definition;
     Weight 25;
     Name FRLINK;
     Arcs
     FRLINK -> FIXER,
     FRLINK -> RECEIVER,
     FIXER -> ORGANIZATION,
     ORGANIZATION -> BUSINESS,
     RECEIVER -> ORGANIZATION;
}

Graph {
     Cast Definition;
     Weight 25;
     Name RWLINK;
     Arcs
     RWLINK -> WHOLESALER,
     RWLINK -> RECEIVER,
     RECEIVER -> ORGANIZATION1,
     WHOLESALER -> ORGANIZATION2,
     ORGANIZATION2 -> BUSINESS,
     ORGANIZATION1 -> BUSINESS;
}
```

In addition to the schemata, the hierarchy is also necessary to define the sub/super-type relationships between the type labels. This is as follows:

```
Hierarchy {
     BOT -> FIXER,
     BOT -> FINANCIER,
     BOT -> LAUNDERER,
     BOT -> RECEIVER,
     BOT -> WHOLESALER,
     BOT -> COURIER,
     BOT -> DISTRIBUTOR,
     BOT -> FRLINK,
     BOT -> RWLINK,
     BOT -> CONSPIRACY,
     BOT -> IMPORTING,
     BOT -> INSURANCE,
     BOT -> BANKING,
     FIXER -> PERSON,
     RECEIVER -> PERSON,
     LAUNDERER -> PERSON,
     FINANCIER -> PERSON,
     WHOLESALER -> PERSON,
     COURIER -> PERSON,
     DISTRIBUTOR -> PERSON,
     FRLINK -> LINK,
     RWLINK -> LINK,
     CONSPIRACY -> LINK,
     INSURANCE -> FINANCE,
     BANKING -> FINANCE,
     IMPORTING -> BUSINESS,
     FINANCE -> BUSINESS,
     ORGANIZATION1 -> ORGANIZATION,
     ORGANIZATION2 -> ORGANIZATION,
     PERSON -> TOP,
     LINK -> TOP,
     ORGANIZATION -> TOP,
     BUSINESS -> TOP;
}
```

'TOP' is the universal type (which has no super-type), and 'BOT' is the absurd type (which has no sub-type).

Error Processing in the Hunch Buddy Demonstration

The current demonstration software can show only a few of the error types previously discussed. Hunch Buddy concentrates on Sp, and to a lesser extent Cl, so the error types that can be demonstrated are focused on errors in hypothesis generation. These errors can spring from a variety of sources:

a. Misinterpretation of information, i.e., the spanning set settles on hypotheses that do not anticipate new facts, or worse, may be incoherent with new facts.

b. Failure to integrate information at the right level of detail, i.e., covers of facts generated by Sp are insufficiently connected, or insufficiently rich in concepts, or insufficiently specialized.

c. The generation of sparse hypotheses that do little more than re-represent available facts, i.e., cover parameters tend to generate sparse structures.

All of these errors can be demonstrated by altering the knowledge base. If schemata are incorrect (as opposed to merely incomplete) then error (a) will occur. If they are incomplete then error (b) will occur. If the schemata are too small (contain too few links and introduce too few new concept types) then error (c) will occur.

In addition to these errors, which are errors in knowledge, the software can demonstrate how factual errors (in the database) can be propagated through to hypotheses, or cause no hypotheses to be generated. For instance if a type is introduced in the data that is not contained in any schema, then no covers will be obtained. If two concepts are linked in a database item that are never linked in any schema, then the linkage will appear in all covering hypotheses, possibly leading to errors later on.

Proposals for Further Work

According to the discussion of intelligence analysis, the key to anticipating a threat's intentions may be the generation an appropriate spanning set of intentions hypotheses over the larger set of physical and doctrinal possibilities. There is as yet little public research either into the nature of effective spanning sets, or into the dynamics of generation. However, e-MGR was originally developed with such research in mind. It is therefore proposed that IPPM/e-MGR work be focused on simulating those errors arising from hypothesis reformulation or refinement based on new information or reassessment of old information (errors IV (4)). The different aspects of the origin of these errors could then be investigated by altering the parameters of the e-MGR mechanisms. All of the hypothesis errors (translated into e-MGR terms) can be simulated parameters to Cl, Sp, and Fr. It is interesting to note that many of them may have a variety of causes. Some of these can be demonstrated in the current software.

a. Misinterpretation of information, i.e., the spanning set settles on hypotheses that do not anticipate new facts, or worse, may be incoherent with new facts.

b. Incomplete use of information, i.e., e-MGR does not sufficiently specialize its hypotheses from a given fact, or fails to pick up a new fact because the relevant portion of the hypothesis has been fragmented away, or Cl weighs assumptions inappropriately.

c. Failure to revise interpretations of facts given new information, i.e., Sp fails to generate new covers that join in new schemata given new facts, or the covers are generated, but not passed on to Fr, or covers are passed on but have the relevant interpretive portions fragmented out again.

d. Failure to integrate information at the right level of detail, i.e., covers of facts generated by Sp are insufficiently connected, or insufficiently rich in concepts, or insufficiently specialized.

e. Failure to integrate information coming from different subject domains, i.e., covers necessary to link facts from different domains are rejected because of the current complexity settings to "cover," or there are inappropriate access restrictions set within the type hierarchy.

f. The generation of sparse hypotheses that do little more than re-represent available facts, i.e., cover parameters tend to generate sparse structures.

g. The generation of overly complex hypotheses containing much unsupported material, i.e., current cover parameter values tend to generate very integrated structures.

h. Failure to preserve critical information, i.e., the effects of an over-active Fr.

i. The preservation of unnecessary information, i.e., the effects of an under-active Fr.

It may be seen that many of the above individual errors can have several causes in e-MGR terms. It is anticipated that such one-to-many relationships will be common in the study of intelligence production mechanisms.

Conclusions

The main conclusion is that e-MGR can provide a suitable set of mechanisms for augmenting the IPPM. Through the e-MGR mechanisms, the theoretical levels around the IPPM, and cause and effect relationships between levels can be clarified. In addition, the etiology of errors (and their decision effects)can be dynamically sketched. Furthermore, e-MGR will provide the

theoretical foundation for developing compensating, or partially compensating, strategies to minimize the negative effects of errors. For example, there is evidence that negative effects of over-assimilation may be avoided by retaining alternative interpretive structures for data, and of over-accommodation by forcing the justification of each interpretation in terms of alternatives.

Work in progress includes that precise mathematical specification of MGR micro-and macro-theories. MGR software products are available in CommonLisp on Symbolics and Sun Workstations. The e-MGR software that forms the basis of the Hunch Buddy is available on Sun workstations and IBM-compatible PC's.

REFERENCES

Burnstein, D. D., Fichtl, T., Landee-Thompson, B., & Thompson, J. (1990). Implementation Guide for Assessing Intelligence Production Effectiveness (ARI Research Product 90-32). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A229 870)

Charniak, E. (12986). Motivation analysis, abductive unification and non-monotonic equality. Artificial Intelligence, 34, 275-295.

Coombs, M. J., & Hartley, R. T. (1987). The MGR algorithm and its application to the generation of explanations for novel events. International Journal of Man-Machine Studies, 27, 679-708.

Coombs, M. J., & Hartley, R. T. (1988). Explaining novel events in process control through model generative reasoning. International Journal of Expert Systems, 1, 89-109.

Coombs, M. J., Pfeiffer, H. D., & Hartley, R. T. (1991). e-MGR: An architecture for symbolic plasticity. International Journal of Man-Machine Studies, to appear.

Coombs, M. J. (1991). The Insurgency Drug Ring Scenario. Working Paper, Knowledge Systems Group, Computing Research Laboratory, NMSU.

de Kleer, J., & Williams, B. C. (1987). Diagnosing multiple faults. Artificial Intelligence, 32, 97-130.

Fields, C., Coombs, M. J., & Hartley, R. T. (1988). MGR: An architecture for problem solving in unstructured task environments. Proceedings of the Third International Symposium on Methodologies for Intelligent Systems, 40-49. Amsterdam: Elsevier.

Hanks, S., & McDermott, D. V. (1985). Temporal reasoning and default logics. Proceedings of AAAI-86, Philadelphia.

Hartley, R. T., & Coombs, M. J. (1988). Conceptual programming: Foundations of problem solving. In J. Sowa, N. Foo, & P. Rao (eds.), Conceptual Graphs for Knowledge Systems. Reading, MA: Addison-Wesley.

Hayes, P. (1985). The second naive physics manifesto. In J. Hobbs, and R. Moore (eds.), Formal Theories of the Commonsense World. Norwood, NJ: Albex.

McCarthy, J. (1980). Circumscription: A non-monotonic inference rule. Artificial Intelligence/fR, 9. 27-40.

McDermott, D. (1987). A critique of pure reason. Computational Intelligence, e, 151-160.

Minsky, M. (1985). The Society of Mind. New York: Simon and Schuster.

Pierce, C. S. (1934). Scientific method. In A. W. Burks (ed.), Collected Papers of Charles Saunders Pierce. Harvard: Harvard University Press.

Reggia, J., Mau, D., & Wang, P. Y. (1985). A formal model of diagnostic inference. I. Problem formulation and decomposition. Information Sciences, 37, 227-256.

Shoham, Y. (188). Chronological ignorance: Experiments in non-monotonic temporal reasoning. Artificial Intelligence, 36, 279-331.

Sowa, J. F. (1984). Conceptual Structures. Reading, MA: Addison-Wesley.

Staw, B. M. (1976). Knee-deep in the big muddy: A study of escalating commitment to a chosen course of action. Organizational Behavior and Human Performance, 16, 27-44.

Tolcott, M. A., Marvin, F. F., & Bresnick, T. A. (1989). Situation assessment and hypothesis testing in an evolving situation (TR 89-3) (Contract #MDA903-86-C-0332). Decision Science Consortium, Inc.